COMPUTACION APLICADA A LA INDUSTRIA DE PROCESOS

CAIP´2005

José Boaventura Cunha
Manuel Cordeiro
Sérgio Leitão
Editores

Actas del 7º Congreso Interamericano
de Computación Aplicada a la
Industria de Procesos, CAIP'2005
Vila Real, Portugal

Vila Real - Portugal
Septiembre de 2005

Computación Aplicada a la Industria de Procesos, CAIP'2005

José Boaventura Cunha, Manuel Cordeiro, Sérgio Leitão (Editores)

ISBN: **972-669-677-1**

Impreso en la Universidad de Tras-os-Montes e Alto Douro, Portugal

Robotic Manipulator Synthesis using a Hierarchical Multi-objective Genetic Algorithm

E. J. Solteiro Pires	J. A. Tenreiro Machado	P. B. de Moura Oliveira
Dep. Eng. Electrotécnica	Dep. Eng. Electrotécnica	CETAV
UTAD University	Inst. Sup. de Eng. do Porto	UTAD University
5000-911 Vila Real	R. Dr Ant. Bernadino Almeida	5000-911 Vila Real
Portugal	4200-072 Porto, Portugal	Portugal
epires@utad.pt	jtm@dee.isep.ipp.pt	oliveira@utad.pt

Abstract

Robotic manipulator synthesis considering the simultaneous optimization of several design objectives is a NP-hard problem. This paper proposes a hierarchical multi-objective genetic algorithm to generate a robot structure and the required manipulating trajectories. The aim is to minimize the trajectory space ripple, the initial and final binary torques, while optimizing the mechanical structure. Simulations results are presented from solving a structure synthesis problem which considers the optimization of three simultaneous objectives.

Introduction

In the last twenty years genetic algorithms (GAs) have been applied in a plethora of fields such as: control, system identification, robotics, planning, scheduling, image processing, pattern recognition and speech recognition [1]. This paper addresses the generation of a robotic manipulator structure and the planning of trajectories, namely in finding a continuos motion that takes the hand from a given starting point up to a desired end position in the workspace.

Several single-objective methods for trajectory planning, collision avoidance and manipulator structure definition have been proposed. A possible approach was proposed by Chocron and Bidaud [2] involving an evolutionary algorithm to perform a task-based design of modular robotic systems. The system aims to determine the base position and an arm that may be built with serially assembled links and joints modules. The optimization design is evaluated with geometric and kinematic performance measures. Han et al. [3] describe a design method of a modular manipulator that uses kinematic equations to determine the robot configuration and, in a second phase, adopts a GA to find its optimal length. Kim and Khosha [4] present the design of a manipulator that is best suited for a given task. The design consists in determining both the trajectory and length of a three degrees of freedom (*dof*) manipulator. Another application was proposed by Gallant and Bourdeau [5] to optimize a RPR structure in order to obtain the maximum workspace and put the singularities points off from the workspace.

Multi-objective techniques using GAs have been increasing in relevance as a research area. In 1989, Goldberg [6] suggested the use of a GA to solve multi-objective problems and since then other investigators have been developing new methods, such as multi-objective genetic algorithm (MOGA) [7], non-dominated sorted genetic algorithm (NSGA) [8] and niched Pareto genetic algorithm (NPGA) [9], among many other variants [10]. Following this line of thought, this paper proposes the use of a multi-objective method to optimize a manipulator trajectory. This proposed method is based on a GA adopting direct kinematics. The optimal structure front is the one that minimizes the objectives.

The rest of the paper is organized as follows: section formulates the problem and the GA based method for its resolution. Section presents several simulations results involving different robots, objectives and workspace settings. Finally, section outlines the main conclusions.

Problem and algorithm formulation

This study considers robotic manipulators that are required to move from an initial point up to a given final position. In the experiments 1 up to 4 dof planar manipulators were adopted with rotational and prismatic joints. The arm link length are in the range $[0.1, 1]$ m with increments of 0.1 m, and the robot rotational joints are free to rotate 2π rad. Therefore, the manipulator workspace is a circle with a 4 m maximum radius. In what concerns the *structure* generator, it is adopted a hierarchical GA, with 3 GAs to search.
The hierarchical EA is adopted in this work with four EAs, see figure 1. A MOEA is used to evaluate the robot's structure, *structure* generator. For each structure population element three single GAs are executed. Two GAs are used to calculate the initial and final configurations of the trajectory. The third GA determines the intermediate configurations between the two points calculated previously, called *trajectory* generator, in order to find an optimal robot path. Therefore, for each structure are executed three GAs and the best fitness for each GA are used to form the three objective values of the structure solution.

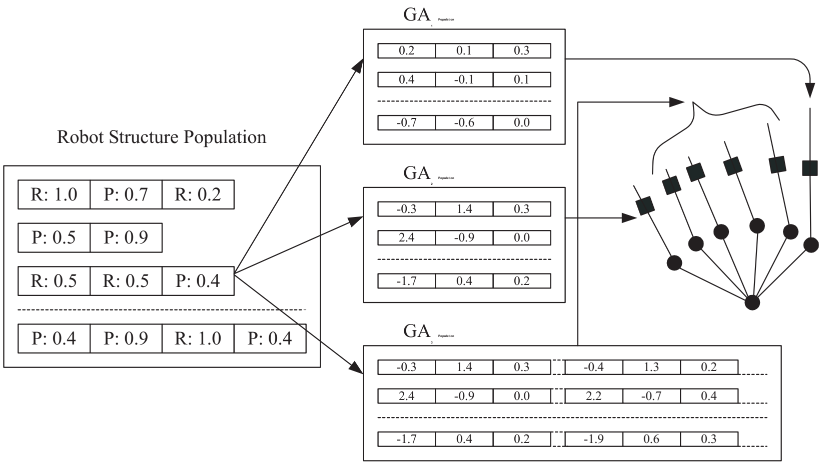


Figure 1: Hierarchical GA

Representation

The robotic structure string is represented in (1) where J_i represents the type of the i^{th} joint (this variable can take two values: R for rotational and P for prismatic joints) and l_i is the i^{th} link length, in the range $[0, 1]$ m with allowed increments of 0.1 m. In order to limit the computational time the number of *dof* is limited to $k \leq 4$. All values used in this work are encoded through real values except the type of the robotic link.

$$S_{\{J:l\}} = \{(J_1^{(T)}, l_1^{(T)}), \ldots, (J_i^{(T)}, l_i^{(T)}), \ldots, (J_k^{(T)}, l_k^{(T)})\} \tag{1}$$

On the other hand, for the initial and final configuration are encoded as (2).

$$\{q_1^{(T)}, \ldots, q_k^{(T)}\} \tag{2}$$

Finally, the path is encoded, directly, as strings in the joint space to be used by the GA as:

$$\{(q_1^{(1,T)}, \ldots, q_k^{(1,T)}), \ldots, (q_1^{(j,T)}, \ldots, q_k^{(j,T)}), \ldots, (q_1^{(n-2,T)}, \ldots, q_k^{(n-2,T)})\} \tag{3}$$

In the generation T, the i^{th} joint variable for a robot intermediate j^{th} position is $q_i^{(j,T)}$, the chromosome is constituted by $n - 2$ genes (configurations) and each gene is formed by k values. The values of $q_i^{(j,0)}$ are initialized in the range $]-2\pi, 2\pi]$ for R-joints and $[0.1, 1]$ m for the case of P-joints. It should be noted that the initial and final configurations have not been encoded into the string because this configuration remains unchanged throughout the trajectory search. Without losing generality, for simplicity, it is adopted a normalized time of $\Delta t = 0.1$ s between two consecutive configurations, because it is always possible to perform a time re-scaling.

Operators in the multi-objective genetic algorithm

The initial populations are generated at random. The search is then carried out among these populations. The different operators used in the *trajectory* planning are reproduction, crossover and mutation, as described in the sequel. Successive generations of new strings are reproduced on the basis of their fitness function. In this case, it is used a rank selection to select the strings from the old population, up to the new population with $\sigma_{\text{share}} = 0.01$ and $\alpha = 2$. To promote population diversity a metric count is used. This metric uses all solutions in the population independently of their rank to evaluate every fitness function. For the crossover operator it is used the simulated binary crossover (*SBX*)[8]. After crossover, the best solutions (among both parents and children) are chosen to form the next population. The mutation operator consists on several actions namely, changing the type of the joint, modifying the link length and changing the joint variable. The mutation operator replaces one gene value with a given probability using equation (4) at generation T, where $N(\mu, \sigma)$ is the normal distribution function with average μ and standard deviation σ.

$$q_i^{(j,T+1)} = q_i^{(j,T)} + \mathrm{N}(0, 1/\sqrt{2\pi}) \tag{4}$$

The operators used for the *structure* optimization are: duplication operator, p_d, that divide one link in two links with same length; the fusion operator, p_r, that join two links; and the mutation operator that changes the length link following equation (5). In all operators the link length restrictions are kept. At the end of each structure GA iteration, the next structure population is selected based on the maximin scheme structure [11].

$$l_i^{(T+1)} = l_i^{(T)} + \mathrm{N}(0, 1/\sqrt{2\pi}) \tag{5}$$

Evolution criteria

Three indices $\{f_{\tau_i}, f_{\tau_f}, f_q\}$ (6) are used to qualify the evolving trajectory robotic manipulators. These criteria are minimized by the planner to find the optimal Pareto front. Before evaluating any solution all the values such that $|q_i^{((j+1)\Delta t,T)} - q_i^{(j\Delta t,T)}| > \pi$ are readjusted, adding or removing a multiple value of 2π, in the strings.

$$f_\tau = g \sum_{j=1}^{k} \sum_{i=j}^{k} m_i \sum_{p=1}^{i-1} l_p(\cos(\theta_p)(j \leq p) + 0.5\cos(\theta_i)) \tag{6a}$$

$$\theta_p = \sum_{i=1}^{p} q_i \tag{6b}$$

$$f_q = \sum_{j=1}^{n} \sum_{l=1}^{k} \left(\dot{q}_l^{(j\Delta t,T)}\right)^2 \tag{6c}$$

The gravitational torque (6a) of extreme positions is used in order to minimize the energy required particularly when the manipulator has long stops points.
The joint distance f_q (6c) is used to minimize the manipulator joints travelling distance. For a function $y = g(x)$ the curve length is defined by:

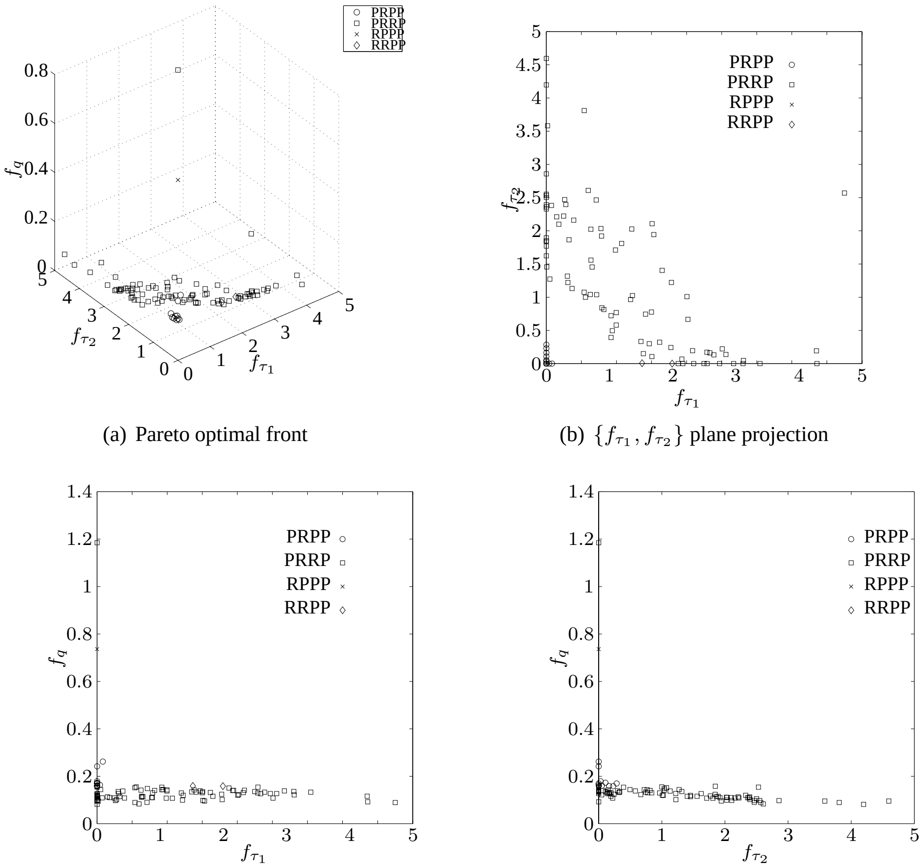
$$\int[1 + (\mathrm{d}g/\mathrm{d}x)^2]\mathrm{d}x \tag{7}$$

and, consequently, to minimize the curve length distance the following simplified expression is adopted:

$$\int(\mathrm{d}g/\mathrm{d}x)^2\mathrm{d}x = \int \dot{g}^2\mathrm{d}x. \tag{8}$$

Simulation results

The experiments consist on moving a robotic arm from the starting point $A \equiv \{1.0, 0.8\}$ up to the final point $B \equiv \{-0.4, 1.2\}$. The simulations results were achieved by using the following GA settings, with $n = 9$ configurations, $T^{(c,t,s)} = \{200, 15000, 1200\}$ for configuration, trajectory and structure generations, respectively. The population size is $pop_{\text{size}}^{(c,t,s)} = \{200, 100, 100\}$, duplication probability $p_d = 0.1$, fusion probability $p_r = 0.1$, crossover probability $p_c = 0.8$ and mutation probability $p_m = 0.1$.



(a) Pareto optimal front

(b) $\{f_{\tau_1}, f_{\tau_2}\}$ plane projection

(c) $\{f_{\tau_1}, f_q\}$ plane projection

(d) $\{f_{\tau_2}, f_q\}$ plane projection

Figure 2: Pareto optimal front $\{f_{\tau_1}, f_{\tau_2}, f_q\}$ and Pareto optimal front plane projections: $\{f_{\tau_1}, f_{\tau_2}\}$, $\{f_{\tau_1}, f_q\}$ and $\{f_{\tau_2}, f_q\}$

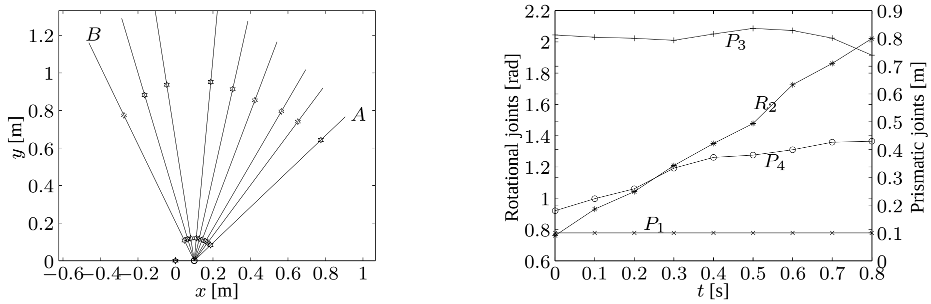
The algorithm determines the non-dominated front maintaining a good distribution of solutions along the Pareto front (figure 2) since the spacing index [12] is SP $= 0.293$ and the Minimal Distance Graph index [11] is MDG $= 0.278$. However, solutions along f_q objective are few relatively to the others objectives because the maximin sorting scheme is used without a scale normalization in all objectives.

The extreme performance solutions of the front are different due to the objectives considered. Between these extreme optimal solutions several others were found, that have a intermediate behavior, and which can be selected according with the importance of each objective. The achieved front structures obtained in the simulation are depicted in table 1, in which P and R means prismatic and rotational joint, respectively. In figure 3 to 6 are shown same different structures of the front. In (a) figures are illustrated the successive configurations of the structures where a circle means a rotational joint and a star represents a prismatic joint. In (b) figures it can be seen the joint position of trajectory *vs.* time where J_i represents the joint type $J = \{\mathrm{R}, \mathrm{P}\}$ for the link $i = \{1, \ldots, 4\}$. The rotational and prismatic scales are in the left and right side of the graphs.

Analyzing the final number of axis, we conclude that the larger the number of dof the better the robot ability to

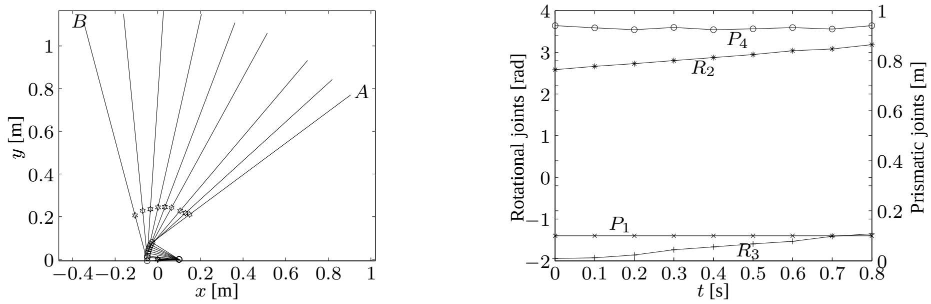
Table 1: Structures Obtained

Structure	Number of solutions
PRPP	10
PRRP	87
RPPP	1
RRPP	2



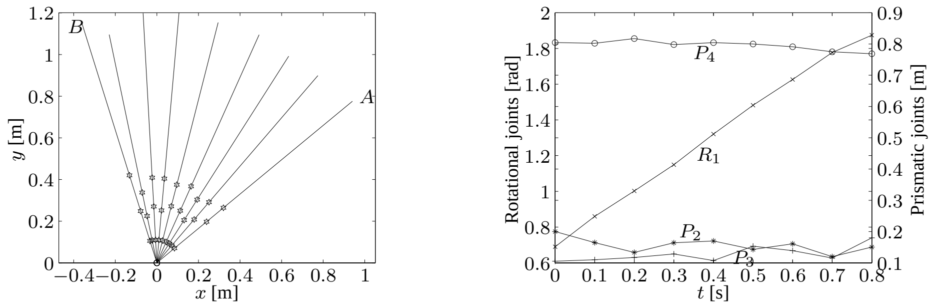
(a) Successive configurations (b) Joint position of trajectory *vs.* time

Figure 3: PRPP robot manipulator



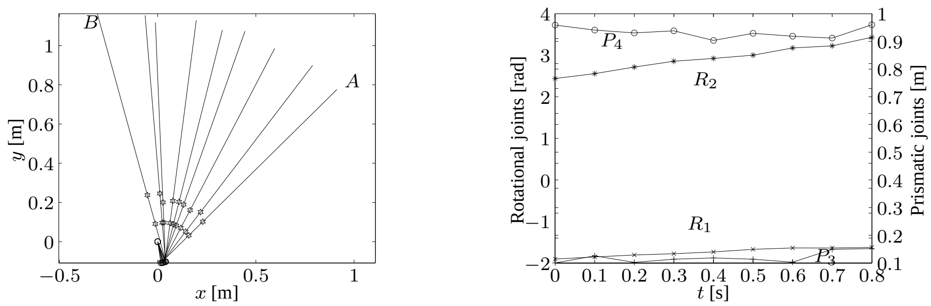
(a) Successive configurations (b) Joint position of trajectory *vs.* time

Figure 4: PRRP robot manipulator



(a) Successive configurations (b) Joint position of trajectory *vs.* time

Figure 5: RPPP robot manipulator



(a) Successive configurations (b) Joint position of trajectory *vs.* time

Figure 6: RRPP robot manipulator

maneuver and to reach the desired points. The structure have a rotational joint near of the robot base. From (b) figures can be seen that the joint cross distance are very near of the optimal.

Summary and conclusions

A multi-objective genetic algorithm robot structure and trajectory planner, based on the kinematics approach, was proposed. The multi-objective genetic algorithm is able to reach optimal solutions regarding the optimization of multiple objectives. The algorithm is able to reach Pareto front and the solutions presents a low gravitational binary at the start and end positions and a reduced ripple in the space trajectory evolution according to objective selected. Furthermore, the algorithm determines the robot structure more adaptable to a given number and type of tasks, maintaining good manipulating performances. Simulation results were presented considering the optimization of three simultaneous objectives.

Acknowledgment

This paper is partially supported by the grant Prodep III (2/5.3/2001) from FSE.

References

[1] T. Bäck, U. Hammel, H.-P. Schwefel, Evolutionary computation: Comments on the history and current state, IEEE Trans. on Evolutionary Computation 1 (1) (1997) 3–17.

[2] O. Chocron, P. Bidaud, Evolutionary algorithms in kinematic design of robotic system, in: IEEE/RSJ International Conference on Intelligent Robotics and Systems, Grenoble, France, 1997, pp. 279–286.

[3] J. Han, W. K. Chung, Y. Youm, S. H. Kim, Task based design of modular robotic manipulator using efficient genetic algorithm, in: IEEE Int. Conf. on Robotics and Automation, New Mexico, USA, 1997, pp. 507–512.

[4] J.-O. Kim, P. K. Khosla, A multi-population genetic algorithm and its application to design of manipulators, in: IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems, Raleight, North Caroline, 1992, pp. 279–286.

[5] M. Gallant, R. Boudreau, The synthesis of planar parallel manipulators with prismatic joints for an optimal, singularity-free space, Journal of Robotic Systems 19 (1) (2000) 13–24.

[6] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[7] C. M. Fonseca, P. J. Fleming, An overview of evolutionary algorithms in multi-objective optimization, Evolutionary Computation Journal 3 (1) (1995) 1–16.

[8] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley-Interscience Series in Systems and Optimization, 2001.

[9] J. Horn, N. Nafploitis, D. Goldberg, A niched pareto genetic algorithm for multi-objective optimization, Proceedings of the First IEEE Conference on Evolutionary Computation, 1994, pp. 82–87.

[10] C. Coello, A. Carlos, A comprehensive survey of evolutionary-based multiobjective optimization techniques, Knowledge and Information Systems 1 (3) (1999) 269–308.

[11] E. J. Solteiro Pires, P. B. de Moura Oliveira, J. A. Tenreiro Machado, Multi-objective maximin sorting scheme, in: Conference on Evolutionary Multi-criterion Optimization – EMO 2005, Springer-Verlag, Lecture Notes in Computer Science Vol. 3410, Guanajuanto, México, 2005, pp. 165–175.

[12] J. R. Schott, Fault tolerant design using single and multicriteria genetic algorithm optimization, Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, Massachusetts (May 1995).